ODD MINIMUM CUT SETS AND b-MATCHINGS REVISITED

ADAM N. LETCHFORD AND DIRK OLIVER THEIS*

ABSTRACT. The famous Padberg–Rao separation algorithm for b-matching polyhedra can be implemented to run in $\mathcal{O}(|V|^2|E|\log(|V|^2/|E|))$ time in the uncapacitated case, and in $\mathcal{O}(|V||E|^2\log(|V|^2/|E|))$ time in the capacitated case. We give a new and simple algorithm for the capacitated case which can be implemented to run in $\mathcal{O}(|V|^2|E|\log(|V|^2/|E|))$ time.

Key Words: matching, polyhedra, separation.

1. Introduction

Let $G = (V, E)$ be an undirected graph, let $b \in \mathbb{Z}_+^V$ be a vector of vertex capacities and let $u \in \mathbb{Z}_+^E$ be a vector of edge capacities. A *u-capacitated b-matching* is a family of edges, possibly containing multiple copies, such that:

- for each $i \in V$, there are at most b_i edges in the family incident on i;
- at most u_e copies of edge e are used.

If we define for each edge e the integer variable x_e, representing the number of times e appears in the matching, then the incidence vectors of u-capacitated b-matchings are the solutions to:

$$\textstyle\sum_{e\in\delta(i)} x_e \le b_i, \quad \text{for all } i \in V \tag{1}$$
$$0 \le x_e \le u_e, \quad \text{for all } e \in E \tag{2}$$
$$x_e \in \mathbb{Z}, \quad \text{for all } e \in E. \tag{3}$$

Here, as usual, $\delta(i)$ represents the set of vertices incident on i.

The convex hull in $\mathbb{R}^E$ of solutions to (1) - (3) is called the *u-capacitated b-matching polytope*. Edmonds and Pulleyblank (see [Edm65] and [Pul73]) gave a complete linear description of this polytope. It is described by the *degree inequalities* (1), the *bounds* (2) and the following *blossom inequalities*:

$$\sum_{e\in E(W)} x_e + \sum_{f\in F} x_f \le \left\lfloor \frac{b(W) + \sum_{f\in F} u_f}{2} \right\rfloor,$$
$$\text{for all } W \subset V,\ F \subset \delta(W) \text{ with } b(W) + \textstyle\sum_{f\in F} u_f \text{ odd.} \tag{4}$$

Here, $E(W)$ (respectively, $\delta(W)$) represents the set of edges with both end-vertices (respectively, exactly one end-vertex) in W, $b(W)$ denotes $\sum_{i\in W} b_i$.

An important special case is where the upper bounds u_e are not present (or, equivalently, $u_{ij} \ge \max\{b_i, b_j\}$ for all $\{i, j\} \in E$). The associated (uncapacitated) b-matching polytope is described by the degree inequalities, the non-negativity

*Supported within project RE 776/9-1 of the Deutsche Forschungsgemeinschaft (DFG).

Algorithm 1 Minimum T-cut [PR82]

Input:
Graph G, set $T \subset V$, and weights $c \in \mathbb{Q}_+^E$.
Output:
A minimum T-cut.

Compute a cut-tree for the graph G with weights c and terminal vertex set T.
For each of the $n-1$ edges of the cut-tree **do**
  Let $\delta(U)$ denote the cut induced by the cut-tree edge.
  Check the cut:
  Compute the parity $|T \cap U| \bmod 2$ and the weight $c(U)$ of the cut.
  If adequate, store U.
End for
Output the best T-cut U.

inequalities $x_e \geq 0$ for all $e \in E$, and the *simplified blossom inequalities*

$$\sum\nolimits_{e \in E(W)} x_e \leq \left\lfloor \frac{b(W)}{2} \right\rfloor, \quad \text{for all } W \subset V \text{ with } b(W) \text{ odd.} \tag{5}$$

In their seminal paper, [PR82] devised a combinatorial, polynomial-time *separation algorithm* for b-matching polytopes. A separation algorithm is a procedure which, given a rational vector $x^* \in \mathbb{Q}^E$ lying outside of the polytope, finds a linear inequality which is valid for the polytope yet violated by x^*. Clearly, testing if a degree inequality or bound is violated can be performed in linear time, so the main contribution of [PR82] is to identify violated blossom inequalities.

For *uncapacitated b-matching,* Padberg & Rao reduce the separation problem to the computation of a minimum T-cut, for which they give a generic algorithm, see Algorithm 1. We will give the definition of the minimum T-cut problem in the next section. Abbreviating $n := |V|$ and $m := |E|$, this algorithm involves the solution of up to $n-1$ maximum flow problems on a graph with $n+1$ vertices and $n+m$ edges. Using the well-known *pre-flow push* algorithm [GT88] to solve the max-flow problems, this leads to an overall running time of $\mathcal{O}(n^2 m \log n^2/m)$.

The Padberg-Rao separation algorithm for *capacitated b-matching,* however, is substantially more time-consuming. It involves the computation of a minimum T-cut on a special graph, the so-called *split graph,* which has up to $n+m+1$ vertices and up to $2m+n$ edges. Up to $n+m-1$ maximum flow problems may be required to be computed. Using the pre-flow push algorithm, this leads to a worst-case running time of $\mathcal{O}(m^3 \log n)$. In 1987, [GH87] observed that the above-mentioned max-flow problems can in fact be carried out on graphs with only $\mathcal{O}(n)$ vertices and $\mathcal{O}(m)$ edges. Although the idea behind this is simple, it reduces the overall running time for the capacitated case to $\mathcal{O}(nm^2 \log n^2/m)$.

In this paper, we propose a new separation algorithm for the capacitated case whose running time is the same as that for the uncapacitated case. As well as being faster than the Padberg-Rao and Grötschel-Holland approaches, the new algorithm is much simpler and easier to implement. It also has a surprisingly simple proof of correctness.

Our results also apply to the case of *perfect capacitated b-matchings.*

Algorithm 2 Blossom minimization

Input:
Graph G, set $T \subset V$, and weights $c, c' \in \mathbb{Q}_+^E$.
Output:
A minimum blossom.

Compute a cut-tree for G with weights $\min(c, c')$ and terminal vertex set V.
For each of the $n-1$ edges of the cut-tree **do**
  Let $\delta(U)$ denote the cut induced by the cut-tree edge.
  Check the cut:
  Compute $\beta(U)$ as in (8).
  If adequate, store U along with the arg-min F.
End for
Output the best blossom (U, F).

As well as being of interest in the context of matching, the algorithm has an important application to the *Traveling Salesman Problem* (TSP). The special blossom inequalities obtained when $b_i = 2$ for all i and $u_e = 1$ for all e are valid for the TSP, and facet-inducing under mild conditions, see [GP79a], [GP79b]. Thus we obtain a faster exact separation algorithm for the TSP as a by-product. In fact, the algorithm is applicable to a general class of cutting planes for integer programs, called $\{0, 1/2\}$-Chvátal-Gomory cuts, see [CF96].

Parts of the contents of this paper appeared in the proceedings of the Xth IPCO conference [LRT04]. However, the proof of correctness of the algorithm is now *substantially* facilitated.

2. Algorithms for minimum T-cut and blossom minimization

Given a graph $G = (V, E)$, an even-cardinality set $T \subset V$ and non-negative rational edge-capacities $c \in \mathbb{Q}_+^E$, the *minimum T-cut problem* asks for an *odd cut* $(U, \complement U)$ (where $\complement U$ is the complement of U in the vertex set) such that the set $U \subset V$ is *T-odd*, i.e., $|T \cap U|$ is an odd number, and which minimizes, subject to this condition, the submodular function

$$U \mapsto c(U) := \sum_{e \in \delta(U)} c_e.$$

In 1982, Padberg & Rao gave the first polynomial-time combinatorial algorithm for computing a minimum T-cut, see Algorithm 1. The key ingredient is the computation of a Gomory-Hu cut-tree [GH61] in step 1. Given a graph $G = (V, E)$, a set $X \subset V$, and non-negative rational vector of edge-capacities $c \in \mathbb{Q}_+^E$, a *cut-tree with terminal vertex set* X for G and c consists of a mapping $\pi\colon V \to X$ with $\pi(x) = x$ for all $x \in X$, and an adjacency relation $\sim$ on the set X. (We adopt the convention that the edges of G will be denoted by xy, and the edges of the cut-tree by $x \sim y$.) The adjacency relation shall make the set of terminal vertices into a tree. An additional condition is required to hold. Deleting an edge $x \sim y$ of the cut-tree partitions the set X into two sets X_x and X_y, and thus defines a cut $(U, \bar{U})$ in G by letting $U := \pi^{-1}(X_x)$ and $\bar{U} := \pi^{-1}(X_y)$. We call this the cut *induced* by the

edge $x \sim y$ of the cut-tree. Now, the condition which is required is the following:

$$\text{for } x, y \in X \text{ with } x \sim y, \text{ the cut induced by this edge of the cut-tree shall be a minimum } (s,t)\text{-cut in } G \text{ with respect to the capacities } c. \tag{6}$$

With the algorithm given by Gomory & Hu, a cut-tree can be computed in time $O(|X| nm \log {n^2}/{m})$.

In Algorithm 1, the time for "checking the cut" in step 4 is negligible (the values $c(U)$ even come for free with the Gomory-Hu algorithm), and hence the Padberg-Rao method for computing a minimum T-cut runs in time $O(|T| nm \log {n^2}/{m})$, as mentioned in the introduction.

Now we come to the blossom separation algorithm of Padberg & Rao [PR82]. Reformulating and generalizing, we say that a *blossom* is a pair (U, F) consisting of a set of vertices $U \subset V$ and a set of edges $F \subset \delta(U)$ with the property that $|T \cap U| + |F|$ is an odd number. Then, if two non-negative rational weight vectors $c, c' \in \mathbb{Q}_+^E$ are given for the edges of G, the blossom separation problem is equivalent to the problem of producing a blossom whose *value*

$$\beta(U, F) := \sum_{e \in \delta(U) \setminus F} c_e + \sum_{f \in F} c'_f$$

is strictly less than, one, if it exists. For the sake of completeness, we describe how this equivalence is established. Padberg & Rao [PR82] introduced, for each $u \in V$, the term $s_u := b_u - \sum_{e \in \delta(i)} x_e$, which is the *slack* of the corresponding degree inequality computed with respect to a given vector x. Then they showed that the blossom inequality (4) can be re-written in the form:

$$\sum_{u \in W} s_u + \sum\nolimits_{e \in \delta(i)} x_e + \sum_{e \in F} (u_e - x_e) \geq 1. \tag{7}$$

To decide if, for a given x, sets W and F exist which violate (7), we define, in a canonical and straight forward manner, a graph G^*, capacities c and c' and a set T of vertices of G^*, in such a way that a blossom with value strictly less than one gives rise to a violated inequality (7) and vice-versa. Let G^* be constructed by adding a new vertex v to $G = (V, E)$ and connecting it with an edge vu to every $u \in V$. Then for each $e \in E$, we let

$$(c_e, c'_e) := \begin{cases} (x_e, u_e - x_e) & \text{if } u_e \text{ is odd} \\ (\min(x_e, u_e - x_e), \infty) & \text{if } u_e \text{ is even} \end{cases}$$

For the edges vu of G^*, we let $c_{uv} := s_u$ and $c'_{uv} := \infty$. Finally, we define T as the set of all vertices u for which the value b_u is odd, and we let $v \in T$ iff $\sum_u b_u$ is odd. Now it is easy to see that for each blossom (U, F) in G^* with $v \notin U$, the inequality (7) with $W := U \cap V$ is violated by $1 - \beta(U, F)$. Note that $\beta(U, F) = \beta(\complement U, F)$.

As mentioned above, the blossom separation Algorithm of Padberg & Rao [PR82] is considerably more complex than the minimum T-cut algorithm. It requires to construct a special graph $\hat{G}$ with $m + n$ vertices and $2m$ edges, on which then a minimum T-cut is computed.

We now give an algorithm for what we call the *blossom minimization problem:* given G, T and c, c' as above, find a blossom (U, F) which minimizes $\beta(U, F)$. The blossom minimization algorithm is displayed as Algorithm 2.

For fixed $U \subset V$, it has been observed by Padberg & Rinaldi [PR90] that

$$\beta(U) := \min\Big\{\beta(U,F) \;\Big|\; F \subset \delta(U),\, |T \cap U| + |F| \text{ odd}\Big\} \tag{8}$$

can be computed in time $O(|\delta(U)|)$ by first tentatively taking $F := \{e \in \delta(U) \mid c'_e < c_e\}$. Now if $|T \cap U| + |F|$ is odd, we have found a minimizing F. Otherwise, find $f \in \delta(U)$ minimizing $|c_f - c'_f|$ over $f \in \delta(U)$, because then the symmetric difference of F and $\{f\}$ minimizes $\beta(U, \cdot)$.

This implies that the loop 2–6 in Algorithm 2 runs in time $O(n^2)$ and that the running time of Algorithm 2 is dominated by the computation of the cut-tree in step 1, which amounts to $O(n^2 m \log {n^2}/{m})$.

The similarity between the Padberg-Rao minimum T-cut Algorithm 1 and our blossom minimization Algorithm 2 is striking. Moreover, in the next section, we give a short and elegant proof of correctness of Algorithm 2, which is similar to a proof of correctness of Algorithm 1 given by Rizzi [Riz02]. At this point, we might note that $\beta(\cdot)$, unlike $c(\cdot)$, is not in general submodular.

3. A simple proof of the correctness of Algorithm 2

Let a cut-tree for G with terminal vertex set X be given, where $X \supset T$. We say that an edge $x \sim y$ of the cut-tree is *T-odd,* if the sets of the bipartition of X defined by $x \sim y$ are T-odd. Thus, the set of T-odd edges of the cut-tree form what is called a T-join, and an edge in the cut-tree induces a T-cut in G if and only if the edge is T-odd. The next theorem is the keystone of the correctness of Algorithm 1. For the sake of clarity, we repeat the proof of [Riz02].

Theorem 3.1 ([PR82])**.** *One of the T-odd edges of the cut-tree induces a minimum T-cut in G.*

Proof. Let U be a minimum T-cut. Now U is a T-odd set, hence there exists an odd number of T-odd cut-tree edges leaving $T \cap U$. Let $x \sim y$ be one of them, and let S be the minimum (x,y)-cut it induces by (6). Since U is an (x,y)-cut, we have $c(S) \le c(U)$, and since $x \sim y$ is an T-odd edge, S defines a minimum T-cut. □

Now we come to the proof of correctness of Algorithm 2.

Theorem 3.2. *One of the edges of of the cut-tree computed in Algorithm 2 induces the a set U which minimizes $\beta(\cdot)$.*

Proof. Let U be a set which minimizes $\beta(\cdot)$. Further, define the set T' as the symmetric difference of T with all sets $\{u,v\}$ for all $e = uv \in E$ $c'_e < c_e$.

Case 1: U is T'-odd. The proof of Theorem 3.1 shows that there exists a T'-odd edge of the cut-tree which induces a minimizer of $\beta(\cdot)$.

Case 2: U is not T'-odd. Let $f = x'y' \in \delta(U)$ have the minimal value of $|c_f - c'_f|$ among all edges in $\delta(U)$. On the path from x' to y' in the cut-tree, at least one edge $x \sim y$ has one end in U and the other not in U. Let S be the minimum (x,y)-cut defined by this edge. Abbreviating $w := \min(c, c')$, we then have

$$\beta(U) = w(U) + \left|c_f - c'_f\right| \ge w(S) + \left|c_f - c'_f\right| \ge \beta(S).$$

The first inequality holds since U is an (x,y)-cut. As for the second, if S is T'-odd, then S minimizes β since $w(S) \le w(S) + |c_f - c'_f| \le \beta(U)$; but if $|T' \cap S|$ is even, then $(S, \{f\})$ is a blossom whence $w(S) + |c_f - c'_f| = \beta(S, \{f\}) \ge \beta(S)$. □

References

[CF96] A. Caprara and M. Fischetti. $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts. *Math. Program.*, 74:221–235, 1996.

[Edm65] J. Edmonds. Maximum matching and a polyhedron with 0-1 vertices. *J. Res. Nat. Bur. Standards*, 69B:125–130, 1965.

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *J. Soc. Ind. Appl. Math.*, 9:551–570, 1961.

[GH87] M. Grötschel and O. Holland. A cutting plane algorithm for minimum perfect 2-matching. *Computing*, 39:327–344, 1987.

[GP79a] M. Grötschel and M. W. Padberg. On the Symmetric Travelling Salesman Problem I: inequalities. *Math. Program.*, 16:265–280, 1979.

[GP79b] M. Grötschel and M. W. Padberg. On the Symmetric Travelling Salesman Problem II: lifting theorems and facets. *Math. Program.*, 16:281–302, 1979.

[GT88] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *J. ACM.*, 35:921–940, 1988.

[LRT04] A. N. Letchford, G. Reinelt, and D. O. Theis. A faster exact separation algorithm for blossom inequalities. In D. Bienstock and G. Nemhauser, editors, *Integer Programming and Combinatorial Optimization 10*, volume 3064 of *LNCS*, pages 196–205. Springer-Verlag Berlin Heidelberg, 2004.

[PR82] M. W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Math. Oper. Res.*, 7(1):67–80, 1982.

[PR90] M. Padberg and G. Rinaldi. Facet identification for the Symmetric Traveling Salesman Polytope. *Math. Program.*, 47:219–257, 1990.

[Pul73] W. R. Pulleyblank. *Faces of matching polyhedra*. PhD thesis, University of Waterloo, 1973.

[Riz02] R. Rizzi. Minimum T-cuts and optimal T-pairings. *Discrete Math.*, 257(1):177–181, 2002.

Adam N. Letchford, Department of Management Science, Lancaster University, Lancaster LA1 4YW, England.
E-mail address: A.N.Letchford@lancaster.ac.uk

Dirk Oliver Theis, Faculty of Mathematics and Computer Science, University of Heidelberg, Germany.
E-mail address: theis@uni-heidelberg.de